

PIZZA SALES ANALYSIS USING SQL

BY: Y Amisha





THE PRIMARY OBJECTIVES OF THIS PROJECT INCLUDE:

- **Sales Analysis:** To identify trends in sales over time, including peak sales periods and seasonal variations.
- **Product Performance:** To evaluate the popularity of different pizza types and toppings, helping to inform menu decisions.
- **Customer Insights:** To analyze customer purchasing behavior, including frequency of orders and average order values.
- **Operational Efficiency:** To assess kitchen performance metrics, such as order preparation times and delivery efficiency.

Retrieve the total number of orders placed

```
select count(order_id) as total_orders from orders;
```

OUTPUT

	total_orders
▶	21350



A close-up photograph of a person's hand holding a wooden spoon and stirring a pan of pasta. The pasta appears to be fettuccine with cherry tomatoes and possibly some meat or vegetables. The pan is dark-colored and sits on a stove. The background is blurred.

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

SELECT

```
ROUND(SUM(order_details.quantity * pizzas.price),  
      2) AS total_sales
```

FROM

order_details

JOIN

```
pizzas ON pizzas.pizza_id = order_details.pizza_id
```

OUTPUT

	total_sales
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA

```
SELECT pizza_types.name, pizzas.price  
FROM pizza_types join pizzas  
ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
ORDER BY pizzas.price DESC  
LIMIT 1;
```

OUTPUT

	name	price
▶	The Greek Pizza	35.95





IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
SELECT pizzas.size,  
COUNT(order_details.order_details_id) AS order_count  
FROM pizzas JOIN order_details  
on pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizzas.size  
ORDER BY order_count DESC  
LIMIT 1;
```

OUTPUT

	size	order_count
▶	L	18526

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

```
SELECT
pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
pizza_types
JOIN
pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN
order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

OUTPUT

	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```
SELECT pizza_types.category,  
       SUM(order_details.quantity) AS quantity  
FROM   pizza_types JOIN pizzas  
        ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
JOIN   order_details  
        ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category  
ORDER BY quantity DESC;
```

OUTPUT

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



GROUP THE ORDERS BY DATE
AND CALCULATE THE AVERAGE
NUMBER OF THE PIZZAS ORDERED
PER DAY

```
SELECT  
    ROUND(avg(quantity), 0) AS avg_pizza_ordered_per_day  
FROM  
    (SELECT  
        orders.order_date, SUM(order_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY orders.order_date) AS order_quantity;
```

OUTPUT

	avg_pizza_ordered_per_day
▶	138



Determine the top 3 most ordered Pizza type based on revenue

```
SELECT pizza_types.name,  
       SUM(order_details.quantity * pizzas.price) AS Revenue  
  from pizza_types JOIN pizzas  
    ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
JOIN order_details  
    ON order_details.pizza_id = pizzas.pizza_id  
 GROUP BY pizza_types.name  
 ORDER BY Revenue DESC  
 LIMIT 3
```

OUTPUT

	name	Revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```
SELECT pizza_types.category,  
ROUND(SUM(order_details.quantity*pizzas.price) / (SELECT  
ROUND(SUM(order_details.quantity*pizzas.price),2)  
AS total_sales  
FROM  
order_details  
JOIN  
pizzas ON pizzas.pizza_id = order_details.pizza_id) *100, 2)  
AS revenue  
FROM pizza_types JOIN pizzas  
ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
JOIN order_details  
ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category  
ORDER BY revenue DESC;
```



OUTPUT

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
SELECT order_date,  
       SUM(revenue) OVER(order by order_date) as cum_revenue  
FROM  
(SELECT orders.order_date,  
       SUM(order_details.quantity * pizzas.price) AS revenue  
FROM order_details JOIN pizzas  
ON order_details.Pizza_id = pizzas.pizza_id  
JOIN orders  
ON orders.order_id = order_details.order_id  
GROUP BY orders.order_date) AS Sales
```

OUTPUT

	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05

THANK YOU



CONTACT

+917090286243

aamisha813@gmail.com

